

# Configuration Management Using Bcfg2

Narayan Desai

Mathematics and Computer Science Division  
Argonne National Laboratory

August 13, 2009



- 1 Background
- 2 Motivating Problems and Goals
- 3 The Bcfg2 Architecture
- 4 Experiences
- 5 Conclusion

## Context

### Bcfg2

- Configuration management tool
- Written in Python
- 18K sloc
- First release in 2004
- Included with most linux distros
- Supports Linux/Solaris/OSX/etc
- BSD licensed

# Background

## Software-Related System Administration Categories

- Software-specific domain knowledge
  - Linux Kernel
  - Apache
  - Globus
- Putting this software together
  - Building coherent services
  - Correctness
  - Robustness
  - Security
  - Fault Tolerance/Fail over

# Background

## Configuration

- Union of software factors that influence the behavior and performance of computer systems

## What do we do with Configuration?

- Creation (Deploy new things)
- Modification (Update existing things)
- Analysis (Acclimation and Troubleshooting)
- Validation (Audits)

## Motivating Problems and Goals

### So, where's the problem?

- Configuration is naturally decentralized
  - Distributed across a large number of devices
- Disorganized
- Inconsistent
- Scaling factors
  - Client count
  - Configuration diversity
  - Number of administrators

# What is needed?

## A representation of goals

- Compact
- Centralized
- Unique
- Accurate
- Verified

## What is needed? (cont.)

### And a process to reconcile goals with reality

- Determine the difference
- Provide change installation on clients
- ... and specification change on the server
- Flexible
- Intuitive

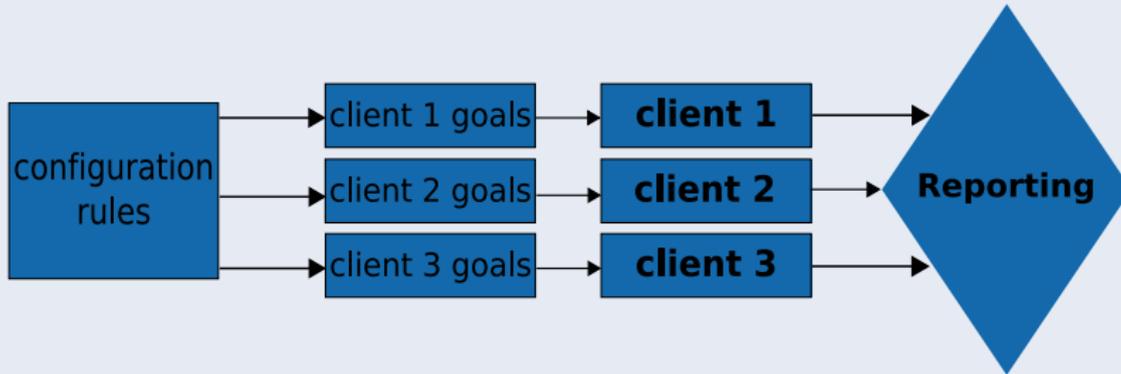
# The Bcfg2 Architecture

## Architectural goals

- Built on verification
- Configuration is discoverable
- Rules used to build per-client configuration goals
- Clients compare goals with current state and reconcile
- Per-client statistics used to construct full system reports

# The Bcfg2 Architecture

## Architectural goals



# The Bcfg2 Architecture

## Configuration goals

- Goals comprised of collections of entries
- Entries correspond to familiar types
  - ConfigFile, Package, Service, etc
  - Verifiable
  - Idempotent
  - Installable
- Comprehensive
- Complete
- Literal
- Typical client goals contain 200-2500 entries

# The Bcfg2 Architecture

## Bcfg2 Client Functions

- Executes local state probes
- Retrieves goals from the server
- Compares current local state to goals
- Determines which goals are unmet
- Attempts to meet those goals
- Report on local state back to the server

# The Bcfg2 Architecture

## Client Execution Triggers

- Init script (boot time)
- Cron jobs (hourly or daily)
- Job prologue/epilogue (in HPC environments)
- Agent mode

# Client Verification Process

## Multi-phase Verification

- Two step process
  - Verify explicit goals
  - Heuristic discovery of *extra* configuration
- Describes goal mismatches
- Discovers (many) unspecified goals
- Gathers all raw data for the reconciliation process

# The Bcfg2 Architecture

## Action Determination

- Normal mode (all pending changes made)
- Dry run mode (no changes made)
- Interactive mode (user prompted for each change)
- Extra entry removal
- Centralized decision mode

## The Bcfg2 Architecture – Statistics

### Client Statistics Data

- Entry counts (total/good/bad)
- Detailed entry information
  - Good Entries
  - Bad Entries
  - Modified Entries
  - Extra Entries
- Detailed Activity information
- Performance data

# The Bcfg2 Architecture

## Reconciliation

- Goals don't always match reality
  - Goals are incomplete or wrong
  - Client configurations are wrong
- Without a convention, it is impossible to know
  - Administrators need to determine the correct course of action
- Tools need to expect this, since reality doesn't always live up to expectations

# The Bcfg2 Architecture

## Reporting System

- A bird's eye view of goal conformance
- Describe reconciliation sites in detail
  - Grouped by pathology
- Multiple output formats (Mail/RSS/Web)
- Provides a good set of metrics for conformance
- *Shows you how you're doing*

# Server Architecture

## The Bcfg2 Server

- Serves data using XML-RPC over HTTPS
- Two main tasks
  - Rendering configuration rules into per-client goals
  - Routing client statistics to the reporting system
- Like all sufficiently mature software, has developed a plugin infrastructure for extension

# Bcfg2 Server Plugins

## Plugin Capabilities

- Configuration goal construction
- Metadata functionality
- Statistics handling
- Configuration validation
- Specialized/customized configuration rule representations

# Bcfg2 Metadata

## Client Metadata

- Describes aspects of the client
  - Group memberships
  - Arbitrary specificity
  - Inheritance
- Includes
  - Client Identifier (hostname or uuid)
  - Group Memberships
  - Probed client information
  - Information from external sources (via Plugins)

## Bcfg2 Server Plugins

### Examples of Stock Plugins

- Cfg
  - Manages configuration files
- Packages
  - Manages package deployment
- SSHbase
  - Manages ssh keys and known\_hosts files
- Templating Plugins
  - Provides a templating interface to generate configuration files. Supports Cheetah, Genshi

## Experiences with bcfg2

### What can you do with bcfg2?

- Push out package installations/upgrades
- Push out configuration file changes
- Pull in manual configuration from clients
- Automate service deployment
  - Services can be configured with templates to automate complex processes

## Experiences

### How has bcfg2 changed things?

- Made us much more efficient
- Provides a useful analogue to documentation
  - Externally verified
  - Automatically updates
  - Close to the mental representation administrators already use
  - Brings new administrators up to speed faster
  - Shared mental model for the Configuration
  - Centralized source of accurate information
- Visibility into the configuration process
  - We know if it is working!

# Conclusion

## Bcfg2 Factoids

- BSD licensed
- Included with most major linux distributions
- Widely deployed across all sectors of IT
- Ran 2 of the top 5 systems on the 11/2008 Top500 list!

Source code, documentation, papers, and much more available at: <http://www.bcfg2.org>

# Conclusion

Questions?